数 値 解 析 37－8
（1991. 7. 17）

条件つきバックプロパゲーション学習法

青山智夫　　市川紘
日立ＣＥ　　星薬科大学

パーセプトロン型ニューラルネットワークの入出力データ間の変換の偏微分値の求め方を提出する．これにより入出力データ間の相関が定量的に計算出来るようになる．　偏微分形を付加条件としてバックプロパゲーション学習に取り入れることにより，ニューラルネットワークのデータ変換作用が学習点だけでなく学習点の近傍まで保証できるようになる．
この条件付きバックプロパゲーション学習はニューラルネットワーク内の，ニューロン間結合の局在化および隠れ層のニューロン数の最小化を行なう再構築学習の方程式に変形できることを示す．

Backpropagation Algorithmn Restricted with some conditions

Tomoo Aoyama, Hiroshi Ichikawa*

Hitachi Computer Engineering Co.Ltd.(Hadano, Kanagawa, 259-13, Japan)
*Hoshi College of Pharmacy.

We have investigated the formula for obtaing partial differentials in the transformation of the perceptron-type neural network. This formulation enables one to correlate inputs with outputs quantitatively. If the formulation is included in the back propagation learning-algorithmn, the transform function which is guaranteed at the learning point is expanded to neighboring regions. The new learning algorithmn can be correlated to the reconstruction learning-algorithmn which localizes the interconnections between neurons and minimizes number of neurons in the hidden layer in the parceptron-type neural network.

1．　はじめに

ニューラルネットワークは複数個のベクトルを異なる個数の異なった要素数からなるベクトルに変換する機能をもっている．特にパーセプトロン型ニューラルネットワークは変換後のベクトルを教師データとしてネットワークに提示し，希望する変換機能を「学習」によって生成する機能がある．機能を実現するアルゴリズムはバックプロパゲーション学習と呼ばれている．この学習方程式によってパーセプトロン型ニューラルネットワークは汎用パターン認識機械として使われるようになった．しかしより広い応用を考えると，次のような問題点が残されている．

1）入出力データ間の相関関係を求める手段が不十分であること，

2）学習点におけるデータ変換作用は保証できるが，それ以外の領域では保証されていない，

3）ニューラルネットワークの構造因子たとえば隠れ層のニューロン数の決定法が未解決のまま残されている．本報告はこれらの問題について一つの解決法を示すものである．

2．　ニューラルネットワークの動作

パーセプトロン型ニューラルネットワークの動作は次のように定義される．

W_{ij}：ニューロン間結合荷重，

x_i：前の層のニューロン出力値（$0 \leqq x_j \leqq 1$），

o_j：ニューロン出力値（$0 \leqq o_j \leqq 1$），

$f(y)$：ニューロン動作関数　とするとき，

$$y_j = \Sigma W_{ij} x_i, \qquad o_j = f(y_j). \tag{1}$$

ニューロン動作関数は一般にはシグモイド関数が用いられる．

$$f(y) = 1 / \{1 + \exp(-y)\} \tag{2}$$

この関数はニューラルネットワークをパターン認識器として用いるときには有用である．　別の目的には線形関数 $f(y) = y$，

ガウス型関数 $f(y) = \exp(-\alpha y^2)$，

不飽和シグモイド型関数 $f(y) = c / \{a + \exp(-y)\}$，where $y \leqq 1$，

ex．（$c = 0.6321206$，$a = 0.2642411$），[2)]

が有効である．異なる種類の関数を組み合わせてネットワークを構成しても良い．[2)]

$-\infty \leqq y \leqq \infty$　区間の関数を使うときバックプロパゲーション（ＢＰ）学習[1)]が使用できる．

$$\delta W_{ij} = -d_j x_i \varepsilon, \tag{3}$$
$$d_j = (o_j - t_j)\, f'(y_j), \tag{4}$$
$$d_j = (\Sigma W_{ij} d'_i)\, f'(y_j). \tag{5}$$

(4)式は出力層のニューロンに関する結合荷重の学習に用いられ，(5)式はそれ以外の層の結合荷重の学習にもちいられる．ただし

t_j：教師データ，

ε：学習係数，

d'_j：学習対象のニューロン層より一つ出力層よりのニューロン層の学習に用いたd_j，

W_{ij}：d'_jを計算したニューロン層間の結合荷重，

$f'(y)$：ニューロン動作関数の微分関数　である．(3~5)のＢＰ学習方程式は複数の入力データ$\{x_j\}_0$について　$E = \Sigma(o_j - t_j)^2$　が十分小さくなるまで繰り返えされる．W_{ij}の初期値は（－１，１）区間の一様乱数である．

$-\infty \leqq y \leqq 1$　のような有限区間の関数を教師データが直接作用しない中間層のニューロンに採用するときはニューロン疲労付きＢＰ学習[3)]を使用することができる．

ＢＰ学習方程式が各$\{x_j\}_0$について$E = \sim 0$となった時，$\{o_j\} = F(\{x_j\}_0)$なる関数Ｆがネットワーク内に生成されたことになる．この関数は｛ｊ｝点で定義されている関数であるが，$o = F(x)$と考えて，$\{x_j\}$に属さないx_uについてもo_uを計算し，これをネットワークによる予測に応用している．

３．偏微分係数

パーセプトロン型ニューラルネットワークの動作は数式によって定義されるから，出力を入力側で偏微分出来る．

各層について，$\delta y_j = W_{ij}\, \delta x_i$，　$\delta o_j = f'(y_j)\, \delta y_j$　(6)

であるから，

２層のネットワークでは，$\delta o_j / \delta x_i = f'(y_j)\, W_{ij}$，　(7)

3層では，$\delta o_J / \delta x_I = \Sigma f'(y_k) V_{ik} g'(z_J) W_{kJ}$ (8)
である．
ただし$f'(y_k)$，$g'(z_J)$はそれぞれ2，3層のニューロンの動作関数の微分関数である．V_{ik}は1，2層間の，W_{kJ}は2，3層間の結合荷重である．

(8)式によって入出力データ間の因果関係を学習済のネットワークから計算できるようになった．

4．結合荷重と偏微分係数の関係

ニューロン間の結合荷重の変化に対して式(8)は次のように変化する．

V_{IK}の微少変化εに対して，$y_K \to y_K + \xi$ と変化するとすると，

$$\xi = x_I \varepsilon,$$

$$f'(y_K + \xi) = f'(y_K) + f''(y_K)\xi \tag{9}$$

ξという変化に対し第2層のニューロンの出力値の変化は，

$f(y_K) + f'(y_K)\xi$． これはW_{KJ}を介して$g'(z_J)$を変化させる．

$$z_J + \eta = W_{KJ}\{f(y_K) + f'(y_K)\xi\},$$

$$\eta = W_{KJ} f'(y_K)\xi = W_{KJ} f'(y_K) x_I \varepsilon,$$

$$g'(z_J + \eta) = g'(z_J) + g''(z_J)\eta \tag{10}$$

故に式(8)のεに対する変化Δ_{IK}は

$$(\delta o_J / \delta x_I) + \Delta_{IK}$$
$$= \sum_{k \neq K} g'(z_J) W_{kJ} f'(y_k) V_{Ik}$$
$$+ [g'(z_J) + g''(z_J)\eta] W_{KJ} [f'(y_K) + f''(y_K)\xi] (V_{IK} + \varepsilon) \tag{11}$$

となる．ここで，

$$\Delta_{IK} = W_{KJ} V_{IK} [g''(z_J)\eta f'(y_K) + g'(z_J) f''(y_K)\xi] + W_{KJ} g'(z_J) f'(y_K)\varepsilon \tag{12}$$

$$\Delta_{IK} = W_{KJ} [V_{IK} [g''(z_J) W_{KJ} \{f'(y_K)\}^2 + g'(z_J) f''(y_K)] x_I + g'(z_J) f'(y_K)] \varepsilon \tag{13}$$

一方，W_{KJ}の微少変化εに対して，$z_J \to z_J + \eta$ と変化するとすると，

$$\eta = f(y_K)\varepsilon,$$

$$g'(z_J + \eta) = g'(z_J) + g''(z_J)\eta$$

$$(\delta o_J / \delta x_I) + \Delta_{KJ}$$
$$= \sum_{k \neq K} g'(z_J) W_{kJ} f'(y_k) V_{Ik}$$
$$+ [g'(z_J) + g''(z_J)\eta](W_{KJ} + \varepsilon) f'(y_K) V_{IK} \tag{14}$$

となる．ここで，

$$\Delta_{KJ} = [g'(z_J)\varepsilon + g''(z_J)\eta W_{KJ}] f'(y_K) V_{IK},$$
$$\Delta_{KJ} = [g'(z_J) + g''(z_J) f(y_K) W_{KJ}] f'(y_K) V_{IK}\varepsilon \tag{15}$$

である．また，

$g(z) = z$　のとき　式(13, 15)は

$$\Delta_{IK} = W_{KJ}\{V_{IK} f''(y_K) x_I + f'(y_K)\}\varepsilon, \tag{16}$$
$$\Delta_{KJ} = f'(y_K) V_{IK}\varepsilon. \tag{17}$$

である．

５．微分ＢＰ学習

入力，教師データの他に偏微分係数が与えられたとき，

$$E = \sum_J (o_J - t_J)^2 + \lambda \sum_{(I,J)} \{(\delta o_J / \delta x_I) - \rho_{IJ}\}^n \tag{18}$$

を結合荷重Ｖ，Ｗで偏微分し，その式を＝0と置いて学習方程式を作ることができる．
ここで，λ，ｎは定数，ρ_{IJ}は偏微分係数の期待値である．
ｎ＝１のとき(18)のλ項は，

$$q_{IJ} = (\delta o_J / \delta x_I) - \rho_{IJ},$$
$$\delta V_{IK} = -\lambda L[\sum q_{IJ} \Delta_{IK}^{-1}\{1 - d(\Delta_{IK})\}], \tag{19}$$
$$\delta W_{KJ} = -\lambda L[\sum q_{IJ} \Delta_{KJ}^{-1}\{1 - d(\Delta_{KJ})\}], \tag{20}$$

である．ここでｄ（ｘ）はｘ＝0のとき１，それ以外のとき0となる関数，
Ｌ［ｘ］は$|x| \geqq \theta$のときsgn（θ，ｘ），それ以外のときｘとなる関数である．

実際の学習は次の様に行なう．

１）ＢＰ学習でＥ～0.001までニューラルネットワークを学習する．

2）式（19, 20）でニューロン間結合荷重を補正する．E値は大きくなる．
定数λ，θはそれぞれ0.001, 0.01程度にとる．
3）Eの値をもとの値よりも僅かに小さくなるように式（3~5）でBP学習を行なう．
4）2，3の操作を$E \to 0$まで繰り返す．Eが小さくならない時λを小さくするが，
$\lambda \to 0$となるときは第2層のニューロンが少ない．

この方法はBP学習による結合荷重の補正部と，(18)式のλ項補正部が独立した操作になっている．

本学習法により学習点の近傍の性質を保証するニューラルネットワークが生成出来，予測機能を向上させることが出来る．付録に数値計算例を示す．

6．再構築学習

BP学習方程式を満たし，入力教師データ間の関係を満足するようなネットワークは何種類も存在する．故に「条件」を式(18)の様に追加して，その条件によってニューロン間の結合を制御することも出来る．　式(18)の代わりに，

$$E_0 = \Sigma\ (o_j - t_j)^2,$$

$$E = E_0 + \lambda \Sigma\ |W_{kl}| \tag{21}$$

を用いると，

$$\delta W_{kl} = -\mathrm{sgn}\ (\lambda,\ W_{kl}), \tag{22}$$

によって結合荷重を修正することになるから，ニューロン間の情報伝達能が絶えず減衰する系を記述したことになる．これは実際のニューロン系に近いモデルである．

ある層のj，k番目のニューロンが同じ種類の情報に関与しており，これらのニューロンに一つ前の層のニューロンiからW_{ij}，W_{ik}という結合を介して情報が伝幡するとする．ただし$W_{ij} > W_{ik}$，　$\mathrm{sgn}\ (1,\ W_{ij}) = \mathrm{sgn}\ (1,\ W_{ik})$．
BP学習によってW_{ij}，W_{ik}がδW_{ij}，δW_{ik}だけ変化する．すると式(3)によって$\delta W_{ij} / \delta W_{ik} = W_{ij} / W_{ik}$．　つぎに式(22)によって$\rho$（$\ll \delta$）だけ$\delta$が減少したとすると，（$\delta W_{ij} - \rho$）／（$\delta W_{ik} - \rho$）$> \delta W_{ij} / \delta W_{ik}$．

情報伝達能の減衰によってW_{ij}，W_{ik}の比は拡大される．この操作がE一定のままで繰り返されると，$|W_{ij}| \gg |W_{ik}|$となって「同種の情報を生成する複数のニューロンが淘汰されて1個のニューロンに集約」される．

式(21)と(18)の添字に関する部分は同じである．従って同様な操作によって式(21)の学習が出来る．

1）ＢＰ学習でE_0～0.1までニューラルネットワークを学習する．
2）式（21）でニューロン間結合荷重を補正する．Ｅ値は大きくなる．
定数λは0.01程度にとる．
3）Ｅの値をもとの値よりも僅かに小さくなるように式（3~5）でＢＰ学習を行なう．
4）２，３の操作をＥ→0まで繰り返す．

この一連の学習を再構築学習と呼ぶ．[4] 本学習により隠れ層のニューロン数を最小にしたニューラルネットワークが生成できる．

７．まとめ

ニューラルネットワークのデータ変換の偏微分式を示し，これによって入出力データ間の相関関係が計算できるようになった．

バックプロパゲーション学習に偏微分係数を期待値に一致させる項や，ニューロン間の結合の絶対値の合計を最小化する項を追加し，微分ＢＰ学習，再構築学習方程式を導いた．これらの学習方程式によって学習点の近傍の性質を保証したネットワーク，ネットワーク内部のニューロン間結合の局在化および隠れ層のニューロン数を自動的に最小化することが出来るようになった．

文献

1) Parallel Distributed Processing Exploration in Microstructure of Cognition; Rumelhart,D.E.,McClelland,N.L.,Eds.;MIT Press:Cambridge,MA, 1986,Vols.1,Chapter 8.
2) T.Aoyama, Y.Suzuki and H.Ichikawa, J.Med.Chem.33,2583(1990).
3) T.Aoyama and H.Ichikawa, Chem.Pharm.Bull. 39,372(1991).
4) T.Aoyama and H.Ichikawa, Chem.Pharm.Bull. accepted for publication.

付録　パーセプトロン型ニューラルネットワークの微分ＢＰ学習計算例

入力データ：ｙ＝ｘ，

教師データ：ｙ＝－3.2＊（ｘ－0.5）²＋Ｒ＊0.05＊Ｓ，　Ｒは（－1,1）区間の乱数，Ｓは（－1）ⁿ 数列，　共に$0<x_i<0.4$で41点等間隔サンプリング．

ニューロン数：　第１層＝２，第２層＝５，第３層＝１．

ニューロンの動作関数：　第２層　sigmoid function,　第３層　linear function.

ネットワークの学習回数：　5000回．

微分ＢＰ学習では１点の偏微分値を合わせるように学習した．この学習はx_i＝0, 0.4の２種類行なった．　計算された偏微分係数と期待値との差は，

i	ＢＰ学習	微分ＢＰ学習(#1)	微分ＢＰ学習(#2)	reference ＢＰ学習(20000回)	
1	-0.0749145	-0.0012136	-0.583276	-0.178307	ｘ＝0.0
2	-0.0421967	0.0177627	-0.494161	-0.136376	
3	-0.0212192	0.0332183	-0.407996	-0.0974888	
4	0.0004938	0.0430136	-0.333026	-0.0671357	
5	0.0149976	0.0500425	-0.261736	-0.0396510	
6	0.0271325	0.0532525	-0.141081	-0.0177214	
7	0.0311403	0.0537676	-0.0934142	0.0003853	
18	-0.0249756	-0.0395848	0.0760692	-0.0046340	
19	-0.0366257	-0.0468033	0.0709243	-0.0117061	
20	-0.0421568	-0.0551996	0.0611488	-0.0207938	
21	-0.0496624	-0.0606791	0.0512419	-0.0279194	
22	-0.0536028	-0.0655605	0.0392655	-0.0350808	
23	-0.0584708	-0.0679019	0.0273258	-0.0404963	
24	-0.0607711	-0.0695008	0.0141820	-0.0455468	
36	0.0494709	0.0571569	-0.0597910	0.0240255	
37	0.0695119	0.0815300	-0.0518153	0.0430227	
38	0.0933305	0.105117	-0.0433560	0.0625839	
39	0.116432	0.131956	-0.0312907	0.0848022	
40	0.144979	0.157855	-0.0187850	0.107557	
41	0.171299	0.189477	-0.0007324	0.134091	ｘ＝0.4
Ｅmax	0.003028	0.003004	0.003406	0.002854	